AD-A040 761 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/2
A STEP MOTOR CONTROLLER FOR A PAN/TILT HEAD.(U)
DEC 76 C C GESCHKE DAAB07-72-C-0259
UNCLASSIFIED R-752 NL

1 OF 1
AD
A040761

END
DATE
FILMED
7-77

REPORT R-752 DECEMBER, 1976 UILU ENG 76-2240

CSL COORDINATED SCIENCE LABORATORY

12 B.S.

A STEP MOTOR CONTROLLER FOR A PAN/TILT HEAD

CLIFFORD CALVIN GESCHKE

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 A STEP MOTOR CONTROLLER FOR A PAN/TILT HEAD.		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER 14 R-752, UILU-ENG-76-224Ø
7. AUTHOR(s) 10 Clifford Calvin/Geschke		8. CONTRACT OR GRANT NUMBER(s) 15 DAAB-Ø7-72-C-Ø259
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS 9 Master's thesis,		12. REPORT DATE 11 Dec[redacted] 1976
		13. NUMBER OF PAGES 46
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 52p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Vision and Image Processing
Real-Time Vision System

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Research into computer vision and image processing is currently being conducted in many places for a great variety of applications. In the Coordinated Science Laboratory at the University of Illinois, emphasis has been placed upon the development of a real-time vision system, that is, one which is capable of interacting with a dynamic environment such as an assembly line in motion, or an active robot manipulator.

LB

097 700

UILU-ENG 76-2240

A STEP MOTOR CONTROLLER
FOR A PAN/TILT HEAD

by

Clifford Calvin Geschke

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-72-C-0259.

A STEP MOTOR CONTROLLER FOR A PAN/TILT HEAD

BY

CLIFFORD CALVIN GESCHKE

B.S., Purdue University, 1973

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1975

Thesis Adviser: Professor Robert T. Chien

Urbana, Illinois

ACKNOWLEDGEMENT

I would like to thank my advisor, Professor Robert T. Chien for guidance and for providing the support and facilities which have made this project possible.

Special thanks also goes to Robert Bales for helping with the mechanical construction, Charles Jacobus and Donald Williamson for listening to countless circuit descriptions, Wesley Snyder for providing encouragement, and Vincent Jones for putting the end result to good use.

TABLE OF CONTENTS

1. INTRODUCTION

Research into computer vision and image processing is currently being conducted in many places for a great variety of applications. In the Coordinated Science Laboratory at the University of Illinois, emphasis has been placed upon the development of a real-time vision system, that is, one which is capable of interacting with a dynamic environment such as an assembly line in motion, or an active robot manipulator.

In order to achieve this goal, a standard vidicon television camera has been interfaced to a DEC PDP-10 computer via an 8-bit A/D converter. Only six of the eight bits may actually be selected at once. The time required to input an entire picture averages about 40 milliseconds. Unfortunately, such speed is paid for in a loss of resolution. A single picture consists of only 238 horizontal lines of 252 samples each.

In order to obtain a higher effective resolution, the TV camera has been equipped with a computer controlled zoom lens assembly which allows a closer look at a smaller portion of the original picture. Obviously, there is a need for a device to move the camera horizontally (pan) and vertically (tilt).

3. MECHANICAL DESIGN

Since it is desirable to aim the camera at a specific point in a scene, the use of a discrete motion device such as a step motor seems natural. Although a DC motor and brake combination could achieve similar results, it would require an analog servo to be interfaced to the controller instead of the TTL compatible step motor driver cards commercially available. The choice of a standard 24 step/revolution step motor driving a 200:1 harmonic drive yields a resolution of 0.075 degrees or 4.5 minutes of arc. This amount of motion corresponds to less than 3 picture cells at maximum zoom. By counting the number of steps traveled from a reference position, an absolute accuracy of 0.1 degrees can be realized. The small amount of error is due to error in sensing the reference position and the spring constant of the harmonic drive unit.

4. CONTROL ALGORITHM

4.1. Basic Step Motor Characteristics

In order to understand the operation of this controller, it is necessary to at least have a superficial understanding of step motor characteristics. Four phase, 24 step/revolution step motors such as those used here can be thought of as discrete motion devices. Each phase has six equally spaced equilibrium points (at 60 degree intervals) arranged so that the four phases make up a total of 24 such points at 15 degree intervals. When a single phase is excited, the motor rotor will tend to move toward the closest of the six equilibrium points associated with it. By exciting the phases in the proper order, the rotor can be made to turn in a desired direction.

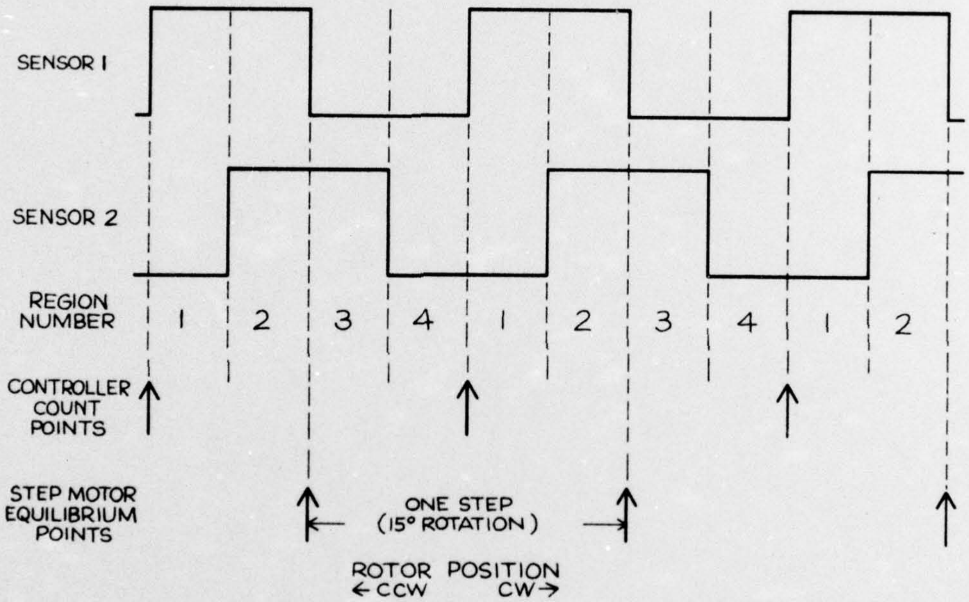
4.2. Motion Sensing

Rotor motion is detected by means of an optical shaft encoder which consists of a thin metal disc with 24 square holes punched along its edge and an LED/photo-transistor sensing device which detects the presence or absence of a hole. Two such sensing devices are placed on the same disc so that the output shown in Figure 1 is obtained when the disc is rotated. As can be seen, the space between possible equilibrium points is divided up into four distinct regions. By noting the order in which the regions occur, the direction of motion as well as the position of the rotor can be determined.

4.3. APHASE Definition

For the purpose of describing the controller, define APHASE to be the relation between the actual rotor position and the excited stator phase.

APHASE = 0 indicates that the rotor is currently at an equilibrium point associated with the excited phase. No net torque results.



OPTICAL SHAFT ENCODER OUTPUT VS. ROTOR POSITION

Figure 1

APHASE = 1 indicates that the rotor is adjacent to the current equilibrium point on the CCW side. A force is exerted to move the rotor in the CW direction.

APHASE = 2 indicates that the rotor is exactly between two equilibrium points. No net torque results, but the position is unstable.

APHASE = 3 indicates that the rotor is adjacent to the current equilibrium point on the CW side. A force is exerted to move the rotor in the CCW direction.

In the following discussion, CW motion is sometimes referred to as positive, forward, or upward, CCW as negative, backward, or downward.

4.4. APHASE Calculation

Clearly, controlling APHASE controls the acceleration of the motor, and allows control of the motor's position. The primary purpose of the controller is to maintain the desired value of APHASE (denoted DPHASE). Since the step motor has only four phases, APHASE is calculated by a modulo 4 up/down counter using the following algorithm:

1) Reset APHASE to 0 when no motion is detected since the rotor must be at an equilibrium point.

2) Increment (decrement) APHASE by 1 when sending a step pulse to excite the next phase in the CW (CCW) direction.

3) Increment (decrement) APHASE by 1 when CCW (CW) rotor motion is detected.

The controller adjusts the value of APHASE by sending step pulses to the motor driver. Step direction is determined according to Table 1.

Table 1. Step Direction Lookup Table

	DPHASE or Bits 3 and 4			
	00	01	10	11
APHASE				
00	--	DN	UP	UP
01	UP	--	DN	UP
10	DN	UP	--	DN
11	DN	DN	UP	--

UP indicates an UP or CW step
DN indicates a DOWN or CCW step
-- indicates no step

4.5. DPHASE Calculation

The actual position (APOS) of either the pan or the tilt axis is computed by counting the actual rotor steps which have occurred since a known reference position. APOS is incremented by one for each CW step, and decremented by one for each CCW step. The relation between APOS and the desired position (DPOS) determines DPHASE:

APOS = DPOS => DPHASE = 0
APOS < DPOS => DPHASE = 1
APOS > DPOS => DPHASE = 3

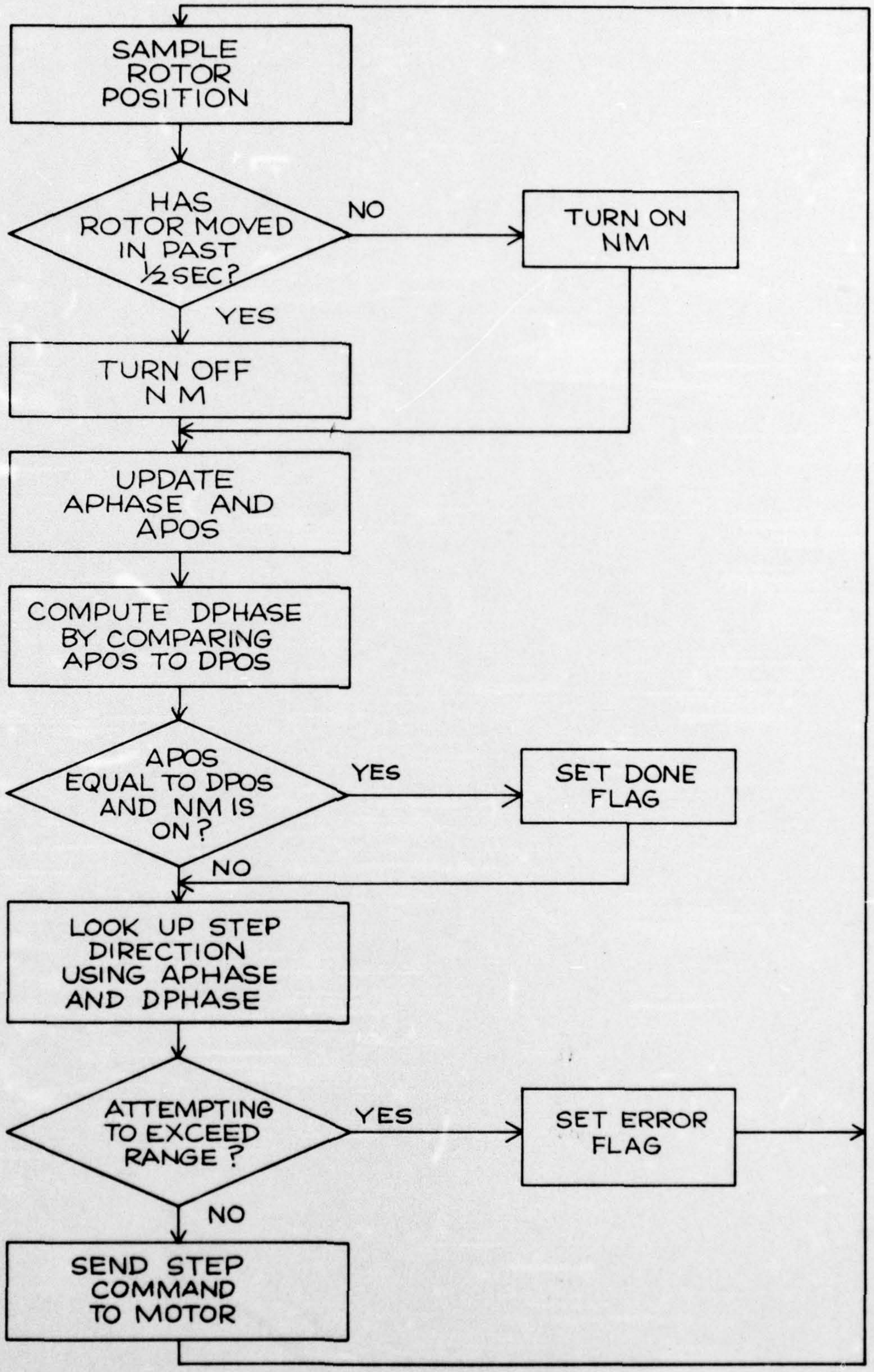
The UP and DOWN outputs of the motion detector are fed to the APHASE counter. No movement (NM) is a signal generated whenever no UP or DOWN signals have been received for one half second.

5. SAMPLE MOTION SEQUENCE

To more fully illustrate the controller operation, a detailed description of the controller states is given during a sample motion sequence. Refer to the Basic Control Algorithm (Figure 2). Assume that the entire system is initially at rest with APOS and DPOS equal to 1000. Therefore NM is asserted, APHASE is 0, DPHASE is 0, and the shaft encoders are producing a 2 or 3 output. Now assume that the computer sets DPOS to 2000. APOS < DPOS means that DPHASE is set to 1 and a forward step is generated to set APHASE to 1. Eventually the rotor will begin to move in the forward direction as a result of the torque which occurs when APHASE is 1. As the shaft encoders pass from region 4 to region 1, the motion detector produces an UP signal which increments APOS to 1001, decrements APHASE to 0, and turns off NM. Since APOS is still less than DPOS, another forward step is generated to restore APHASE to 1.

This procedure is repeated until an UP signal increments APOS to 2000, setting APOS equal to DPOS, while setting APHASE to 0. Since APOS equals DPOS, DPHASE is set to 0, therefore no step pulse is generated. However, the inertia of the mechanical system causes the rotor to keep moving in the absence of applied torque. When the next UP signal is received, APOS is set to 2001, APHASE to 3. APOS greater than DPOS sets DPHASE to 3. Since APHASE is 3, a reverse torque is applied to the rotor, decelerating it. APHASE will be kept at 3 until the rotor stops and reverses direction.

APOS will again reach DPOS, this time from the opposite direction. Further overshoot will set APHASE to 1. Mechanical friction and damping stabilize this procedure. When APOS and DPOS are equal for about one half second, NM is asserted and the motion is considered complete.


SAMPLE ROTOR POSITION
HAS ROTOR MOVED IN PAST ½ SEC?
NO
TURN ON NM
YES
TURN OFF N M
UPDATE APHASE AND APOS
COMPUTE DPHASE BY COMPARING APOS TO DPOS
APOS EQUAL TO DPOS AND NM IS ON?
YES
SET DONE FLAG
NO
LOOK UP STEP DIRECTION USING APHASE AND DPHASE
ATTEMPTING TO EXCEED RANGE?
YES
SET ERROR FLAG
NO
SEND STEP COMMAND TO MOTOR

BASIC CONTROL ALGORITHM

Figure 2

6. MODES OF CONTROL

Several additional modes of control, somewhat different from those described in the previous sections have been implemented to provide a maximum amount of flexibility. They are described briefly below. For complete details, see Section 10, Users Manual.

Local Mode - Inhibits all controller step generation but activates the joysticks to allow manual positioning of the camera. APOS and APHASE are still calculated.

Stop Mode - Inhibits all controller generation.

Seek Mode - Allows the computer to set DPOS. Uses complete control algorithm.

Slew Mode - Allows the computer to set DPHASE. APOS and DPOS have no effect upon step generation.

Single Step Mode - Allows the computer to generate step pulses directly. DPHASE and APHASE have no effect.

Load Mode - Allows the computer to set APOS for initialization purposes.

7. MULTI-PHASE CLOCK DESIGN APPROACH

The control algorithm has been implemented with TTL logic components, using a synchronous multi-phase design approach to cyclically execute the steps shown in Figure 2. A synchronous controller which accepts the output of the shaft encoders and produces appropriate step CW or step CCW signals was designed using a multi-phase clock. As used here, the term "multi-phase clock" refers to a number of signal lines which are sequentially driven by non-overlapping periodic pulses.

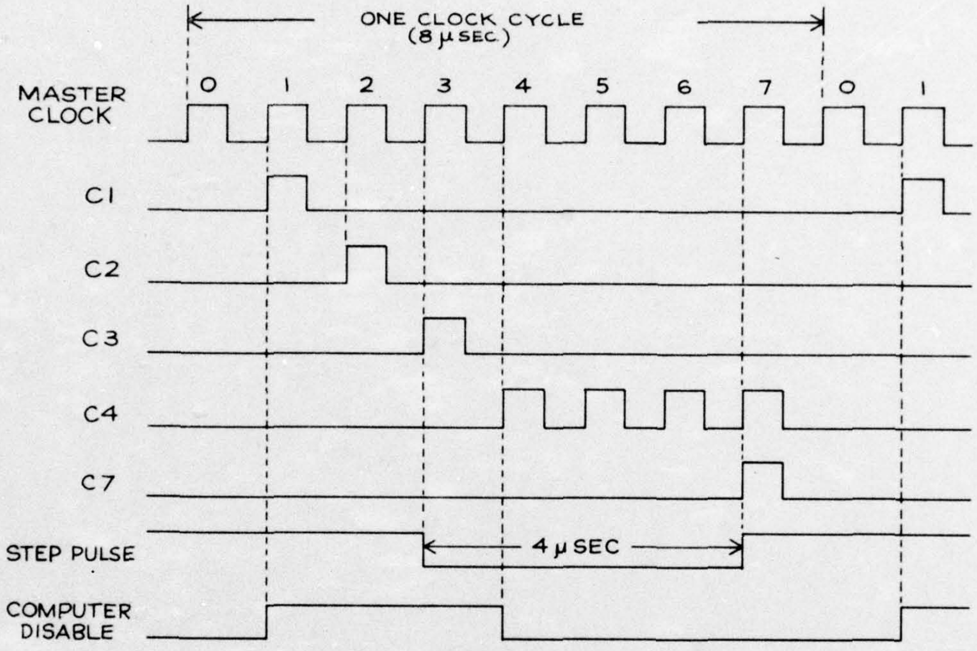
This clock runs at a frequency much higher than the desired stepping rate. In this case, the master clock frequency from which the various phases are derived is 1 MHz and the number of phases is eight, which introduces a worst case delay of 16 microseconds in the control algorithm. Synchronous operation eliminates the need to worry about race conditions since there is sufficient time between phases for the logic gates to settle.

During a single clock cycle, the signals C1, C2, C3, C4, and C7 are sequentially asserted. Refer to Figure 3 for an illustration of relative signal timings. The controller activity during each clock phase is described below.

7.1. Clock Phase 1

All asynchronous signals, that is, all signals which may change independently of the clock, are latched so that they may be treated as static for the duration of the current clock cycle. These signals are:

E1 and E2, the current shaft encoder values;



CLOCK SIGNALS

Figure 3

+ERR and -ERR, the states of the range soft limit switches;

NM, the output of the no movement detector; and

LINE, the value of the LINE/LOCAL switch.

Also, the PDP-11 is inhibited from reading or writing the controller status register (STREG) and position register (PREG). If the NM flag is high, indicating no movement for the past one half second, the rotor is assumed to be stopped, and APHASE is set to 0, indicating an equilibrium condition. The actual position, APOS, is set to the value of PREG if the controller is in load mode (LDM).

7.2. Clock Phase 2

The current values of the shaft encoders (E1 and E2) are compared to the previous shaft encoder values (PE1 and PE2), which were saved during the previous cycle, to determine the current rotor position and direction of travel. A count up signal (CUP) is generated when a count point is passed in the CW direction, and a count down signal (CDN) is generated when a count point is passed in the CCW direction. The count points are located half way between the step motor equilibrium points as shown in Figure 1. Therefore, CUP occurs upon a transition from region 4 to region 1, and CDN occurs upon a transition from region 1 to region 4. If a CUP or CDN signal occurs, the rotor is moving from one equilibrium point to another. CUP causes the value of APOS to be incremented by one, indicating that the rotor has advanced, and it causes the modulo 4 counter containing APHASE to be decremented by one, indicating the change in the net torque on the rotor. In a like manner, CDN causes APOS to be decremented and APHASE to be incremented modulo 4. If either CUP or CDN occurs, NM is reset and begins timing for another one half second.

7.3. Clock Phase 3

By the time phase 3 occurs, the counter containing APOS, the comparator which compares APOS to DPOS, the logic which determines DPHASE from the comparator outputs, and the logic which determines step generation from STREG, DPHASE, or the joysticks, have all reached steady state. A motor step is initiated according to the current controller mode as follows:

Local Mode - Step CW if JUP has been set by the joystick clock, step CCW if JDN has been set by the joystick clock. If either step is generated, both JUP and JDN are cleared.

Seek Mode - Step CW or step CCW is determined by comparing APHASE to DPHASE according to Table 1.

Slew Mode - Step CW or step CCW is determined by comparing APHASE to bits 3 and 4 of STREG according to Table 1.

Single Step Mode - Step CW is initiated if bit 4 of STREG is set. Step CCW is initiated if bit 3 of STREG is set. Both bits 3 and 4 are cleared if either step is generated.

Stop Mode or Load Mode - No steps are initiated.

If a +ERR or -ERR is detected, steps in the CW or CCW direction, respectively, are inhibited regardless of mode. The high byte of STREG is also loaded with the current device status bits at this time.

7.4. Clock Phase 4

The input half of the position register (PREG) is loaded with the current value of APOS. Also, the PDP-11 is enabled to read and write PREG and STREG.

7.5. Clock Phase 7

If a step pulse was initiated during phase 3, it is now terminated, producing a 4 microsecond pulse to the motor driver. Also, the current encoder values are saved as PE1 and PE2 for use in the next clock cycle.

7.6. Unused Clock Phases

The remaining clock phases are unused, but are present so that the computer may have sufficient time to access the command register and the position register.

7.7. Computer Simulation

An additional feature of using a synchronous multi-phase design approach is that the control algorithm can be easily simulated on a minicomputer. Since the sequential nature of the multi-phase clock is analogous to the sequential flow of a computer program, a given sequential algorithm can be tested without the need to construct elaborate special purpose hardware. The control algorithm described previously was programmed on a PDP-11/40 using two binary digits in an input register to read the shaft encoder values and a simple command register to send either a step CW or step CCW pulse to the step motor driver. The simulation routine was programmed in assembly language, and predicted a motor speed of 200 steps/second, no instability problems, and high accuracy. Because of its success, a modified version of the simulation routine was used to control the pan/tilt head while the actual controller was being constructed. A copy of the simulation routine is found in Appendix A.

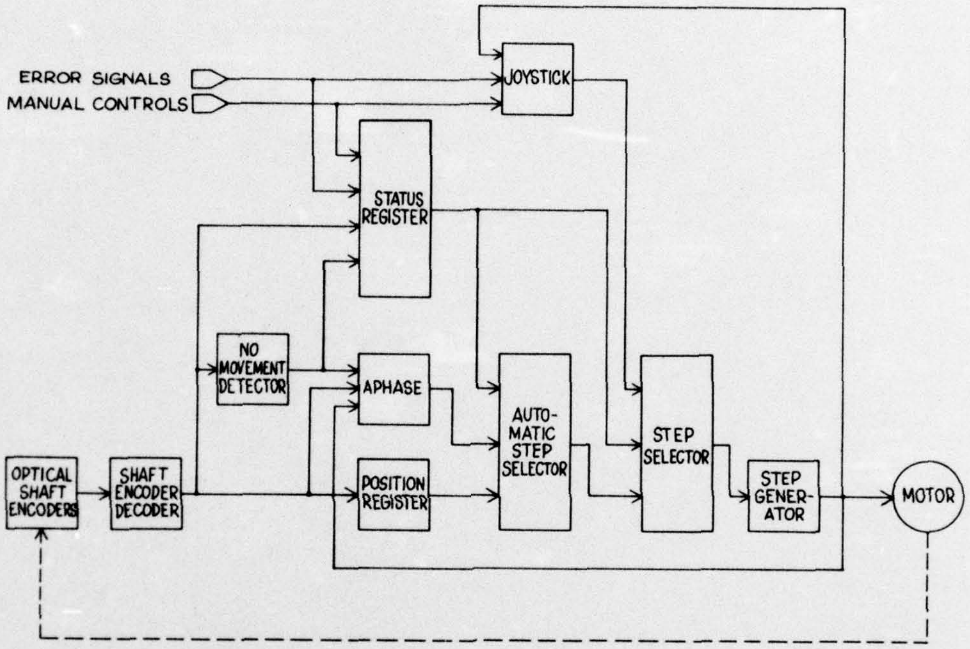
8. CIRCUIT DESCRIPTION

The following sections contain a detailed circuit description of all major circuit blocks, accompanied by schematics. Figure 4, the General Block Diagram shows how these circuit blocks are interconnected. Individual signals are labeled and marked by off-page connectors.

All logic gates and components are standard TTL except for the NE555 timers. For clarity, those logic elements whose functions are completely identified by their symbols have not been labeled. All unspecified inputs are assumed to be at logic level 1 except for the SN74173's whose unspecified inputs are tied to logic level 0.

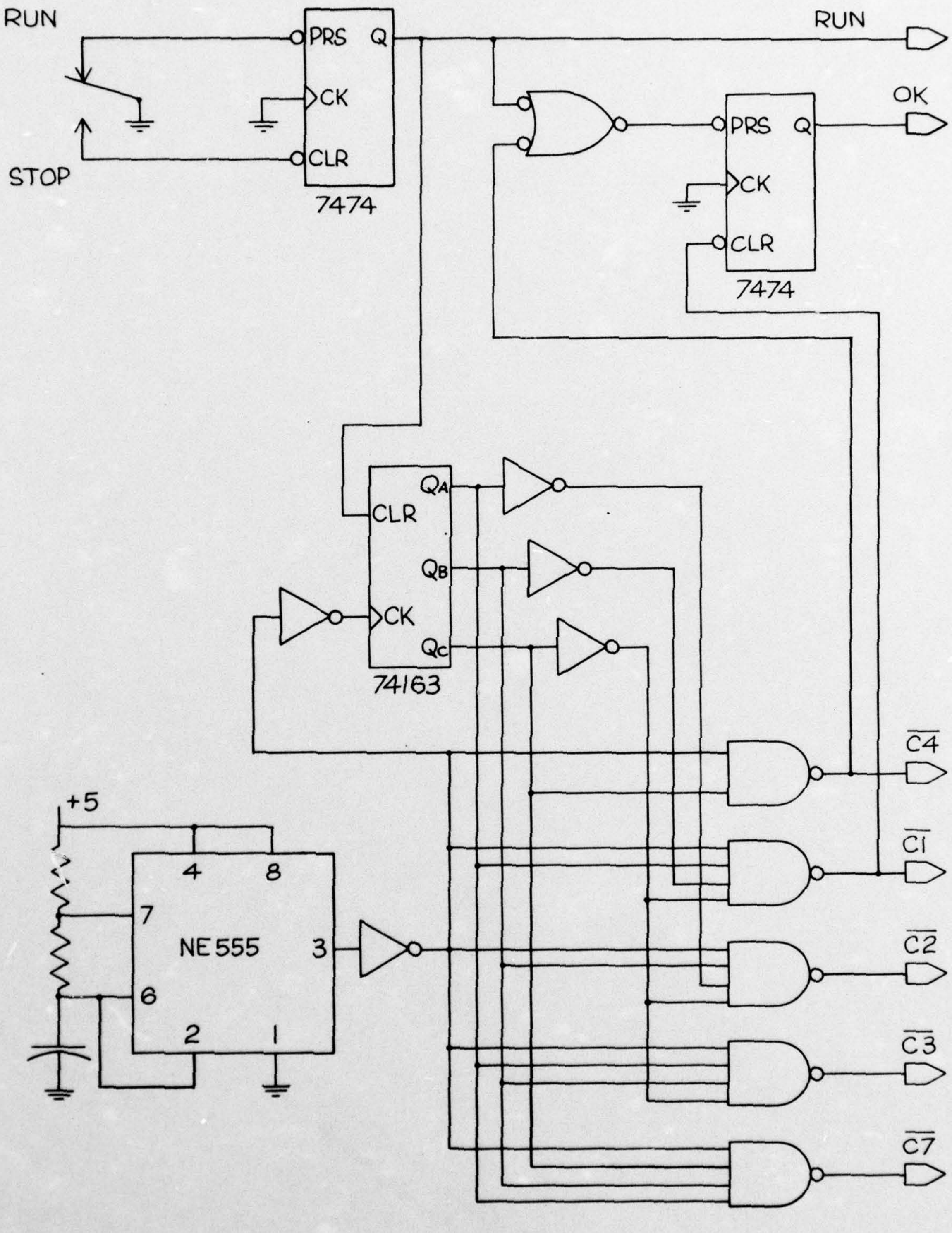
8.1. Multi-Phase Clock - Figure 5

The master clock signal is generated by an NE555 timer connected for a stable operation and buffered by an inverter. This signal is then inverted again and used to clock an SN74163 four bit counter. The three low order bits are decoded to produce the signals 'C1, 'C2, 'C3, 'C4, and 'C7 as shown in Figure 3. A STOP/RUN switch is debounced and used to hold the counter at zero while in the STOP position. The PDP-11 is allowed to read and write STREG and PREG while OK is high. OK is set high at time C4 or when the STOP/RUN switch is placed in the STOP position. OK is reset at time C1 to prevent the PDP-11 from interfering with the controller.



GENERAL BLOCK DIAGRAM

Figure 4



MULTI-PHASE CLOCK

Figure 5

8.2. Joystick Operation - Figures 6a and 6b

Mechanically, each joystick axis consists of two ganged 500K potentiometers and two cam actuated microswitches arranged as shown in Figure 6a. In the center position, both switches are in a state such that both 'SJUP and 'JDN are high. Displacing the joystick from the center actuates one of the switches, depending upon the direction of displacement, and causes either 'SJDN or 'SJUP to fall. The potentiometers on the axis are incorporated into the timing circuit for the NE555 which serves as a joystick clock. They are arranged so that the net resistance of an axis is at a maximum when the joystick is centered and a minimum when displaced to the extreme of either side. Therefore, the clock rate increases as the joystick is displaced from center, providing velocity control. This clock signal is "anded" with SJUP and SJDN to provide directional information. If the step motor is at the limit of its mechanical range in a particular direction, +ERR or -ERR will be high, effectively blocking the joystick signal for that direction. Otherwise, a joystick step request is made by raising JUP or JDN. This request is recognized at time C3 if the controller is in LOCAL mode and is cleared by the generation of any step pulse via 'CLRJ. The JDONE switch and the LINE/LOCAL switch are debounced and passed on to other parts of the controller.

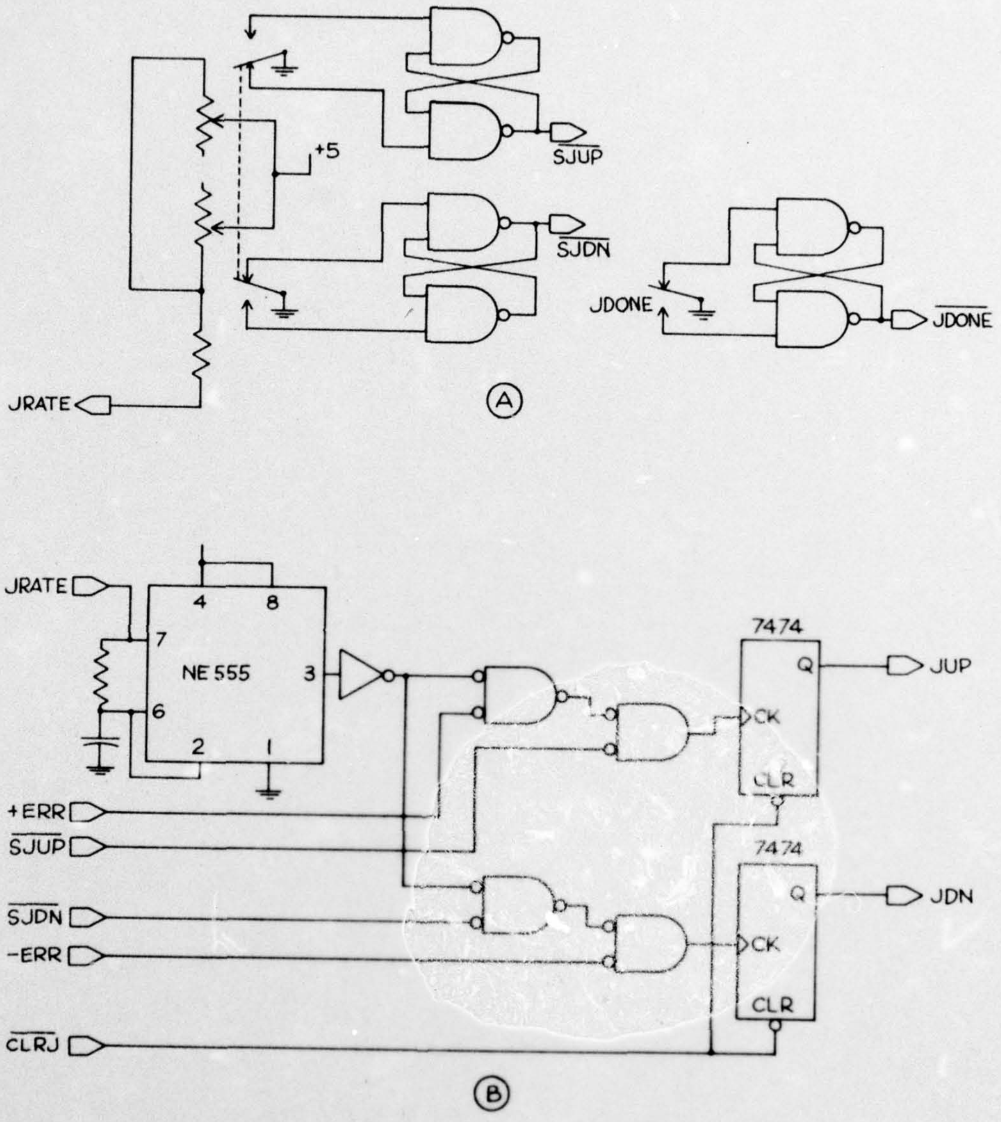

+5
SJUP
SJDN
JDONE
JDONE
JRATE
A
JRATE
4
8
7
6
NE 555
3
2
1
7474
Q
JUP
CK
CLR
+ERR
SJUP
SJDN
-ERR
7474
Q
JDN
CK
CLR
CLRJ
B

Figure 6

8.3. Shaft-Encoder Decoder - Figure 7

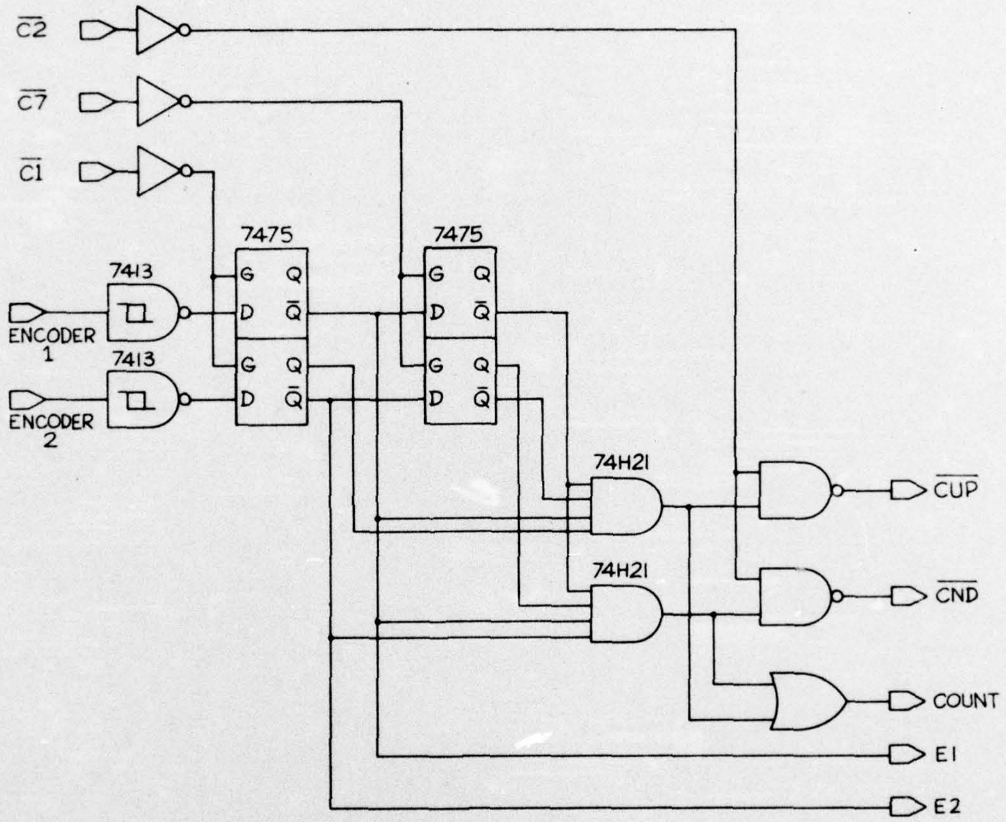
The output of the optical shaft encoders on the rotor shaft is decoded into count up (CUP) and count down (CDN) pulses for use in keeping track of actual rotor position. The output of the encoders is buffered by SN7413 Schmitt triggers, to insure a high degree of noise immunity, and latched at time C1 so that it may be treated as static during the remainder of the control algorithm. The output of these latches is passed on to STREG and is also compared with the previous encoder values via a pair of 4 input AND gates which are sampled at time C2 to generate CUP or CDN. At time C7, the previous encoder values are replaced by the new encoder values for use in the next clock cycle. COUNT, which is valid between times C1 and C7, signals that a CUP or CDN pulse has been generated during the current clock cycle.

8.4. No Movement Detector - Figures 8a and 8b

No movement is signaled whenever an SN74123 retriggerable monostable is not triggered by a CUP of CDN pulse for a period of one half second. This signal, as well as three external signals (LINE, +ERR, and -ERR) are latched at time C1 so that they can be treated as static for the duration of the clock cycle.

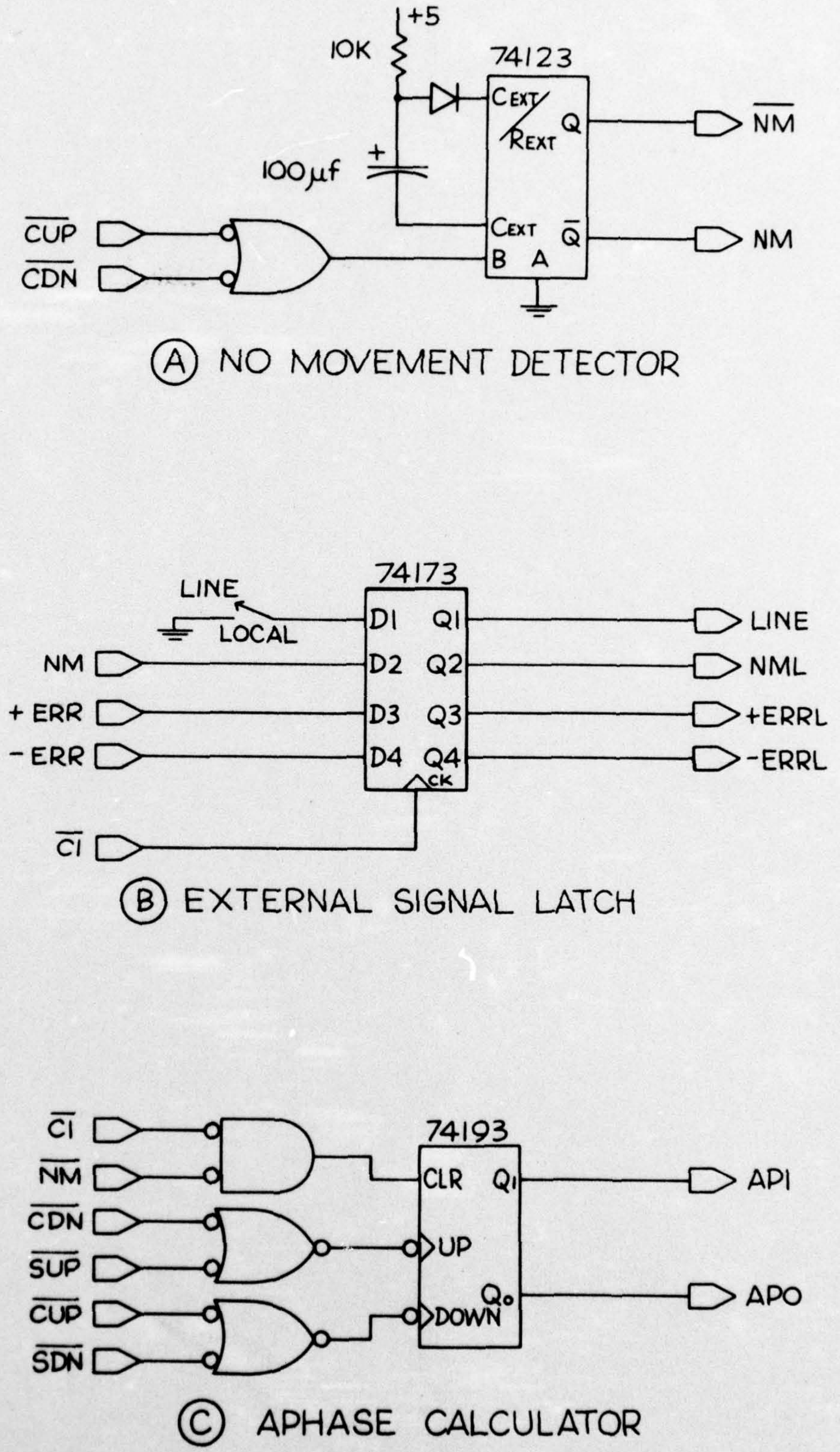
8.5. APHASE Calculator - Figure 8c

The actual phase of the rotor with respect to the energized motor winding is calculated by an SN74193 up/down counter. Since only the two low order bits are used, APHASE is calculated modulo 4 which corresponds to the



SHAFT-ENCODER DECODER

Figure 7



Ⓐ NO MOVEMENT DETECTOR

Ⓑ EXTERNAL SIGNAL LATCH

Ⓒ APHASE CALCULATOR

Figure 8

four phases of the step motor being controlled. If NM is high at time C1, the motor is assumed to be at equilibrium and APHASE is set to zero by clearing the counter. A CDN or SUP pulse causes the counter to be incremented, and a CUP or SDN pulse causes the counter to be decremented. The two bits of APHASE, AP0 and AP1, are passed on to the Automatic Step Selector.

8.6. Automatic Step Selector - Figure 9

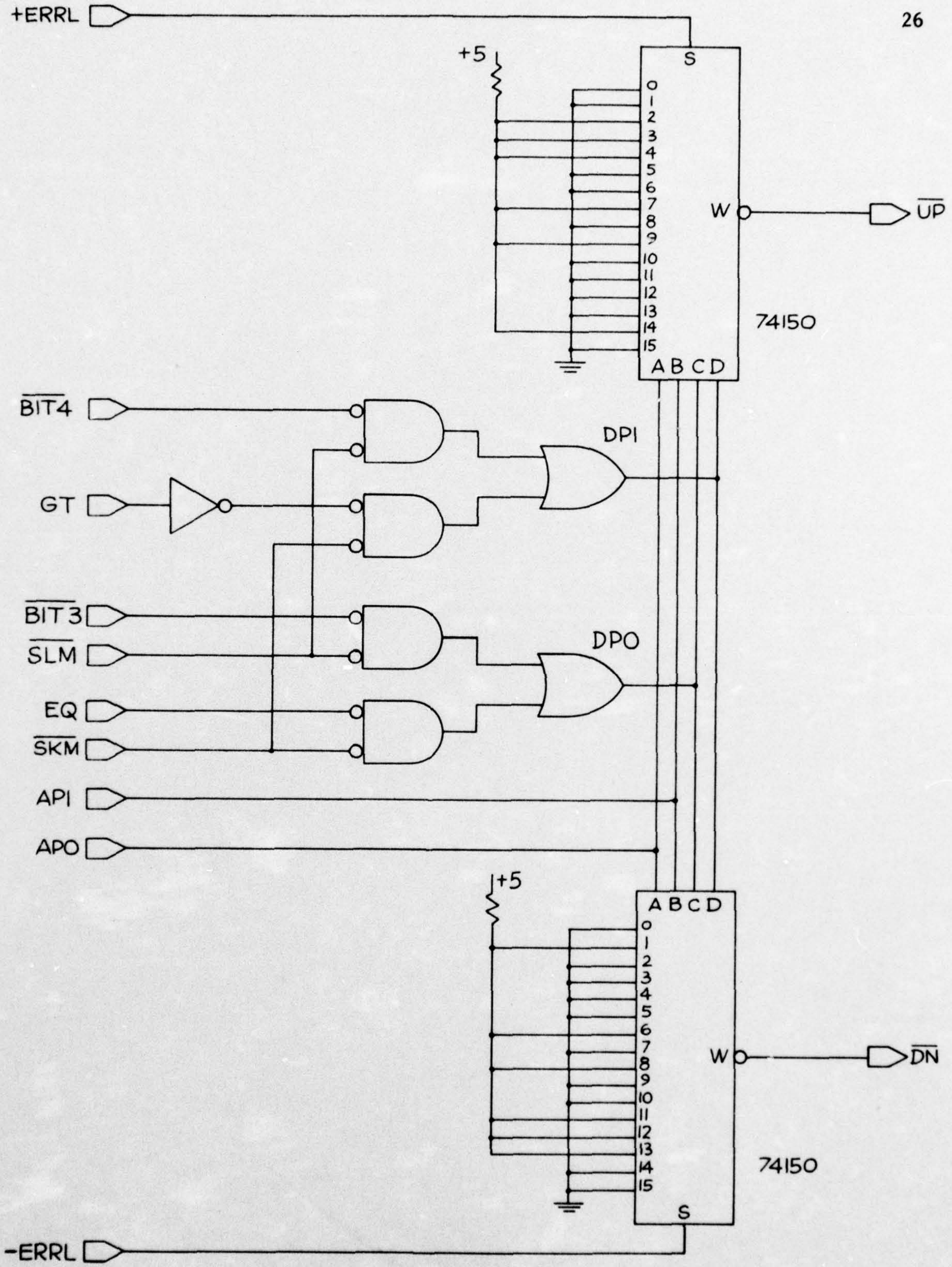
Automatic step selection is accomplished by comparing APHASE (consisting of AP0 and AP1) to DPHASE (consisting of DP0 and DP1) using a table lookup scheme which makes use of two SN74150 16 to 1 multiplexers. The four bits AP0, AP1, DP0, and DP1 are used to address the inputs of the multiplexers and produce UP or DN signals according to Table 1. A +ERR inhibits UP signal generation, and a -ERR inhibits DN signal generation. DPHASE is taken from the APOS to DPOS comparator outputs (EQ and GT) when in Seek Mode (SKM) and from bits 3 and 4 of STREG when in Slew Mode (SLM).

8.7. Step Selectors - Figure 10

Step pulse initiation signals ('IUP and 'IDN) are generated by JUP and JDN when not in LINE mode, by bits 4 and 3 of STREG when in Single Step Mode (SSM), or by UP and DN of the Automatic Step Selector when in Seek Mode or Slew Mode (SKM or SLM).

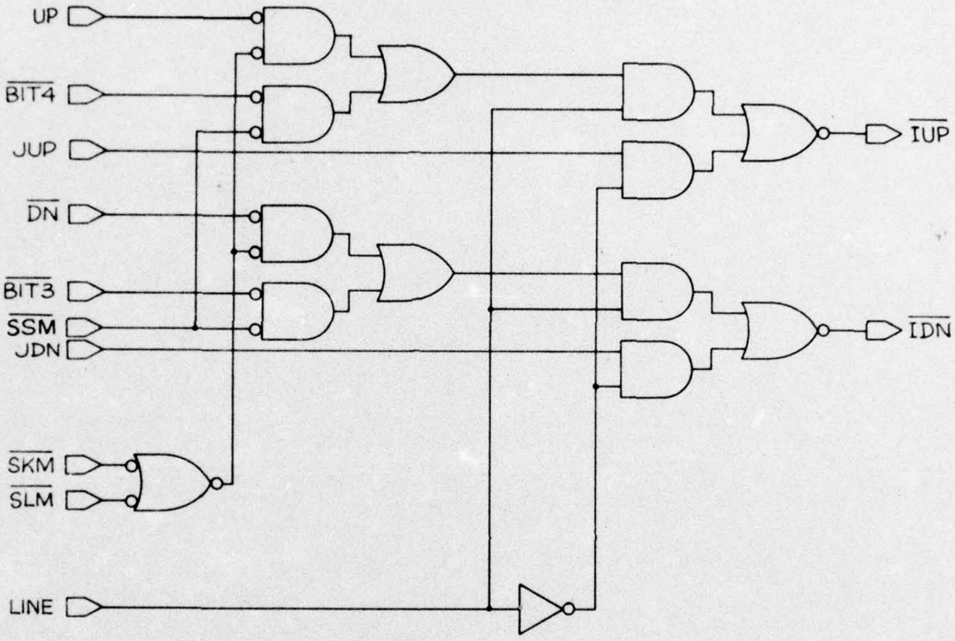
8.8. Step Generator - Figure 11

A step up or step down signal is initiated at time C3 depending upon the value of IUP and IDN which are used to set the step generator



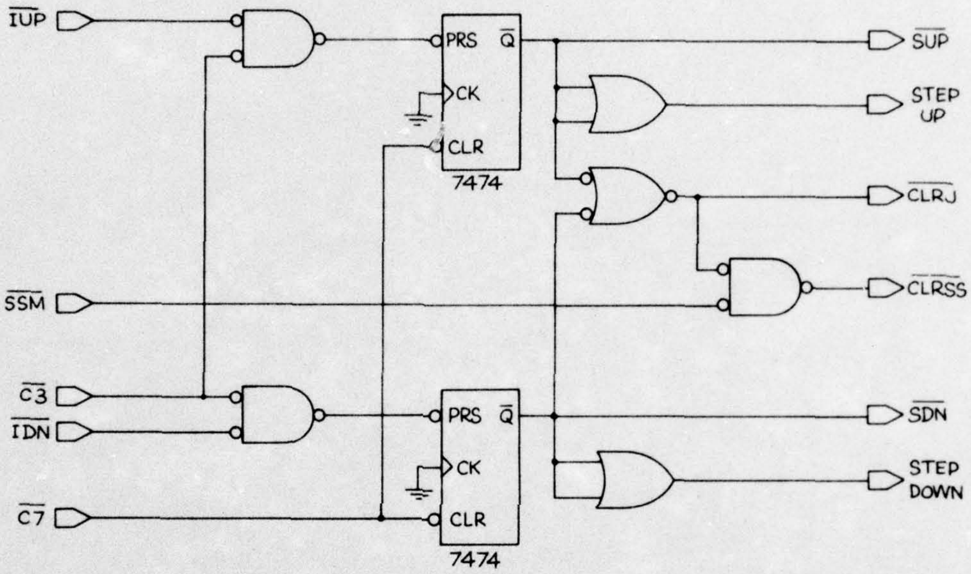
AUTOMATIC STEP SELECTOR

Figure 9



STEP SELECTORS

Figure 10



STEP GENERATOR

Figure 11

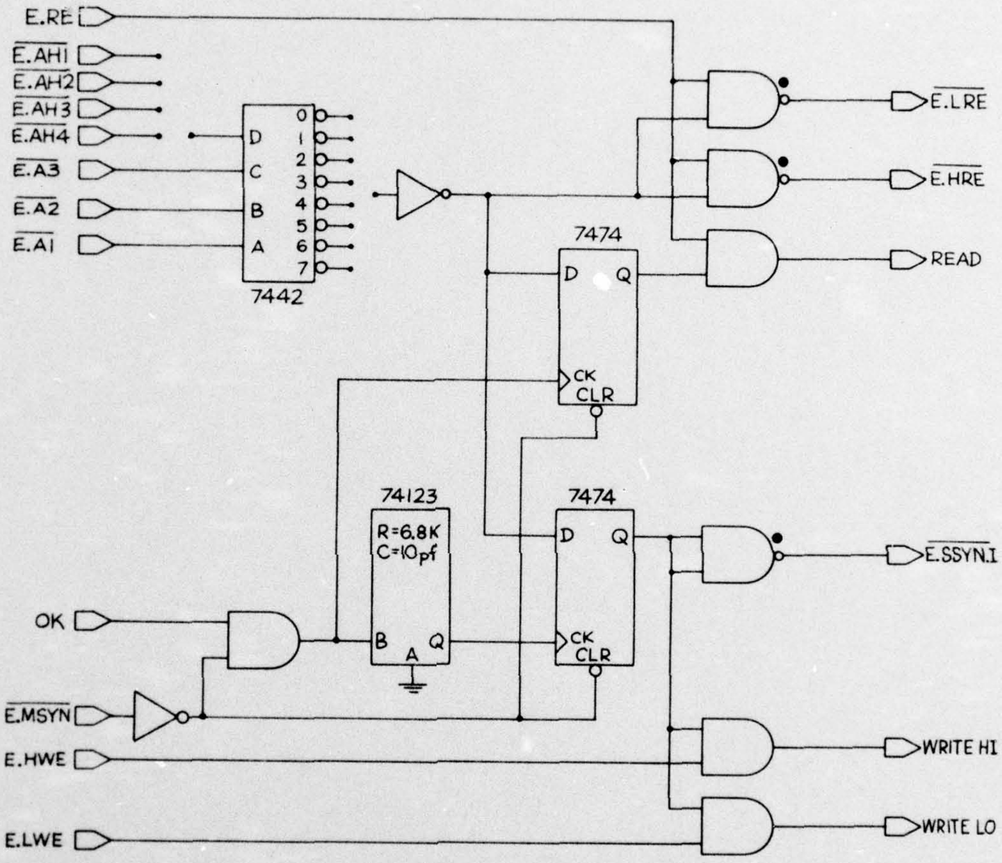
flip flops. These flip flops are cleared at time C7, thus producing a four microsecond pulse suitable for input to the motor driver cards. The signals 'SUP and 'SDN are used in APHASE calculation and are buffered by OR gates to drive the lines to the motor driver. 'CLRJ is the logical "or" of 'SUP and 'SDN and is used to clear any joystick step request. If the controller is in Single Step Mode (SSM), the signal 'CLRSS clears bits 3 and 4 of STREG.

8.9. Status Register

The entire step motor controller is designed to be interfaced to a PDP-11 via a CSL PDP-11 Unibus Expander. Specifications and interfacing conventions may be found elsewhere [3]. The Unibus Expander simplifies address decoding and interrupt requesting and transmits data over a bi-directional tri-state bus denoted as E.D0 through E.D15.

8.9.1. Address Decoder - Figure 12

Only bits 1 through 3 of the address are actually decoded in this portion of the interface. The Unibus Expander decodes bits 4 through 17 and bit 0, and signals the controller via lines 'E.AH1 through 'E.AH4. One of these high address lines is jumpered to the D input of an SN7442, and address bits 1, 2, and 3 are connected to inputs A, B, and C respectively. The actual address is then selected by jumpering to one of the SN7442 outputs. When the OK line is high, enabling the PDP-11 to read and write the status register, a request to read the selected address is signaled by the line



ADDRESS DECODER

Figure 12

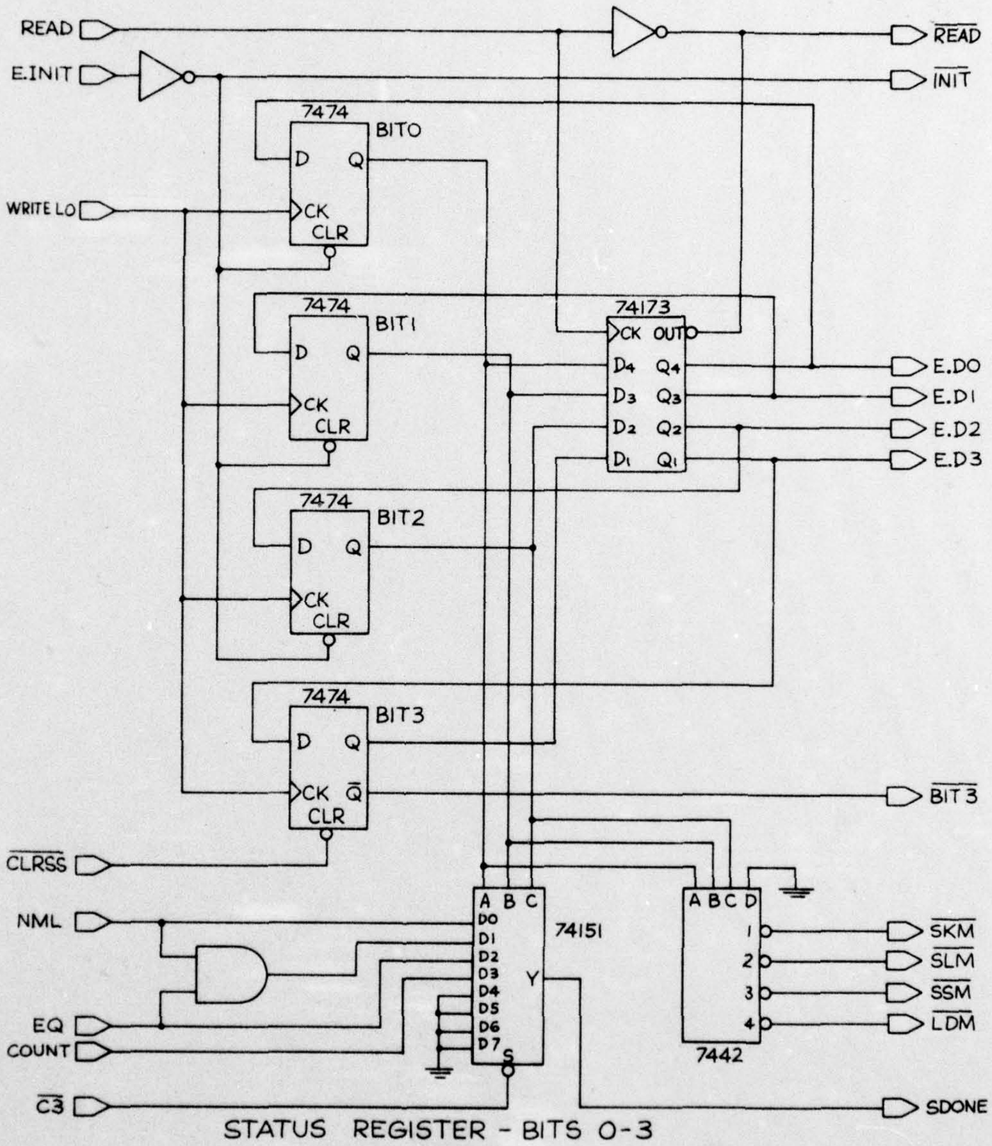
READ being set high, and a request to write the high or low byte is signaled by WRITE HI and WRITE LO respectively. READ causes the tri-state drivers of the status register to be enabled onto the Expanded Unibus data lines. WRITE HI or WRITE LO causes the high or low order byte of the status register to latch the value currently on the Expanded Unibus. All normal PDP-11 Unibus transactions are handled by the Expanded Unibus in conjunction with the Address Decoder.

8.9.2. Status Register, Bits 0 - 3 - Figure 13

Bits 0, 1, and 2 of the Status Register select the controller mode (see Section 6). The current mode is determined by the output of the SN7442 which generates the signals 'SKM, 'SLM, 'SSM, and 'LDM. These bits also determine the done condition by selecting an input of the SN74151 at time C3, and using it to set the DONE flag (bit 7) via the signal SDONE. A device reset clears bits 0, 1, and 2 via E.INIT, placing the controller in STOP mode. Bit 3 is used elsewhere for DPHASE determination in Slew Mode and STEP DOWN generation in Single Step Mode.

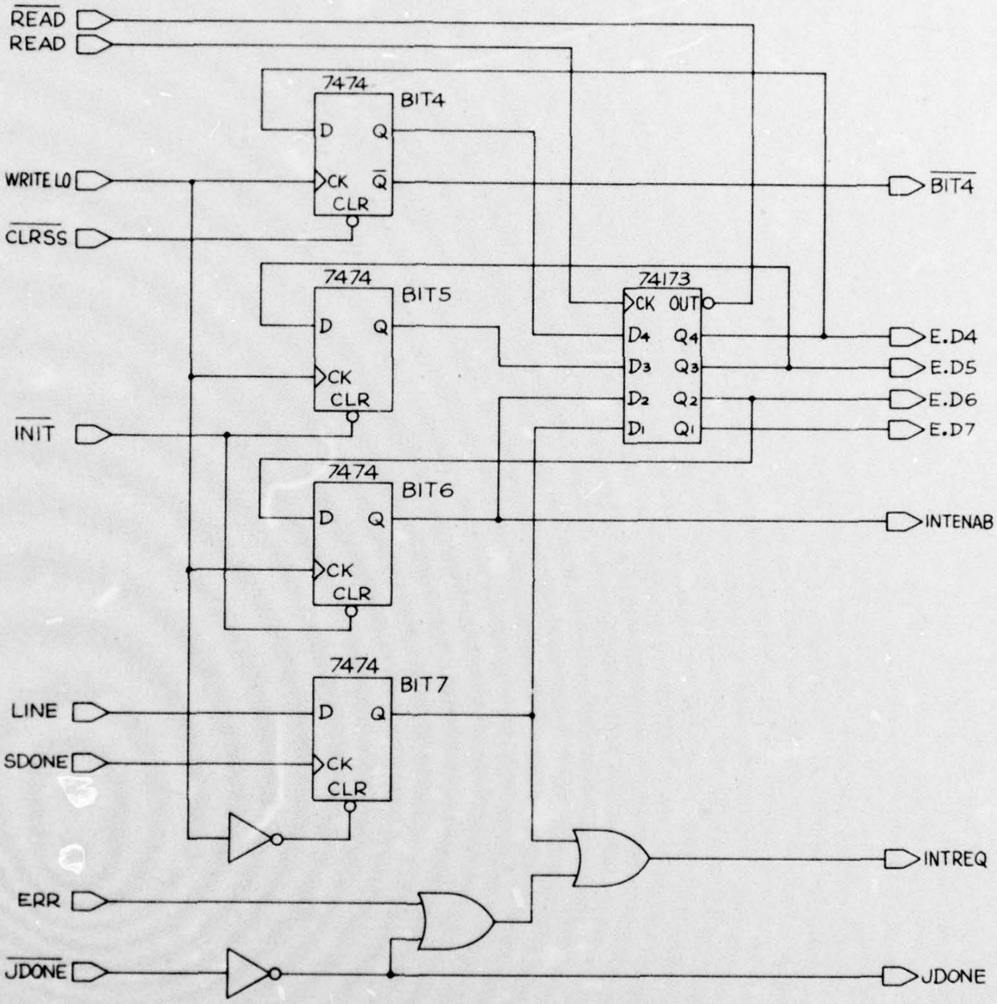
8.9.3. Status Register, Bits 4 - 7 - Figure 14

Bit 4, like bit 3, is used elsewhere. Bit 5 is not used. Bit 6 is the interrupt enable bit which allows the controller to request a processor interrupt when it is set. Bit 7 is the DONE bit which signals that the controller has finished a function sequence. This bit is set by SDONE when the controller is on line. It is cleared by writing the low order byte



STATUS REGISTER - BITS 0-3

Figure 13



STATUS REGISTER - BITS 4-7

Figure 14

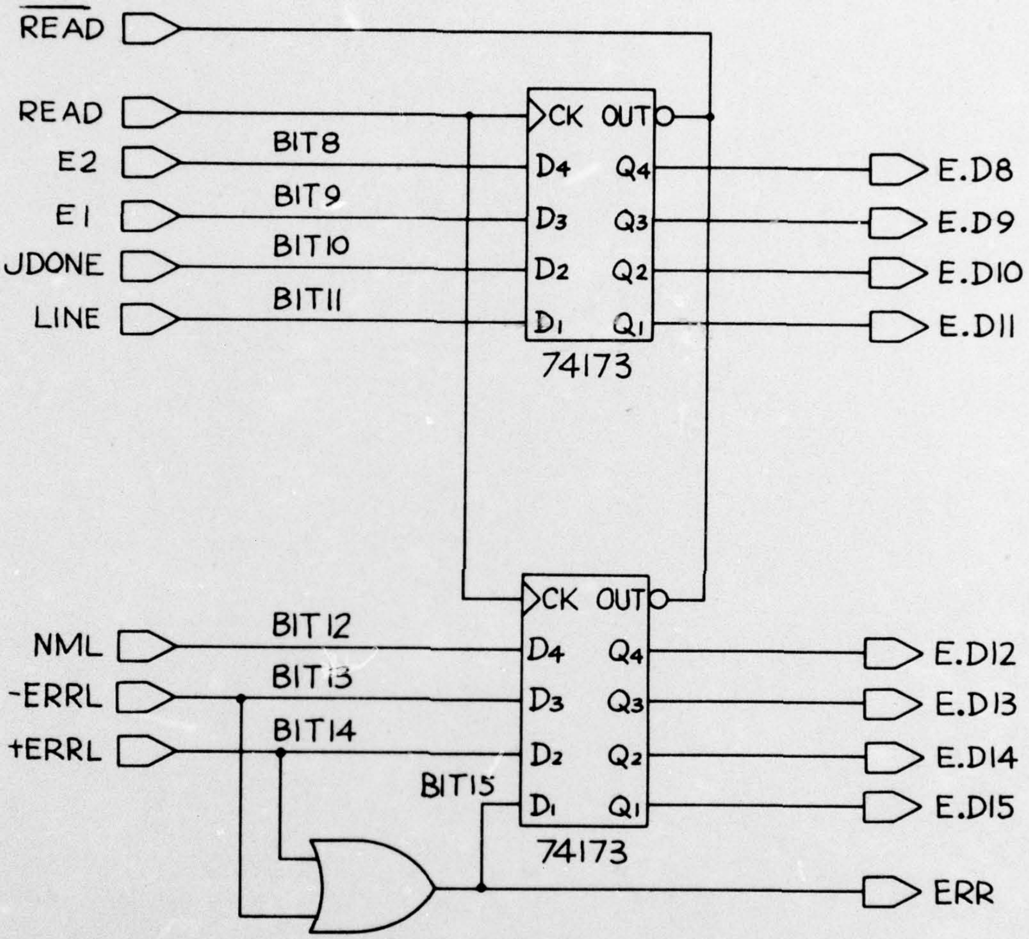
of the register. An interrupt is requested whenever DONE goes high, a limit error occurs, or the JDONE switch is depressed. An interrupt is generated by an Expanded Unibus Interrupt Request Card by the logical "and" of the signals INTENAB and INTREQ.

8.9.4. Status Register, Bits 8 - 15 - Figure 15

The high byte of the status register is read only. It contains information concerning the shaft encoder states and the states of the JDONE and LINE/LOCAL switches, the no movement detector, and the range limit switches. A READ causes this information to be driven to the Expanded Unibus.

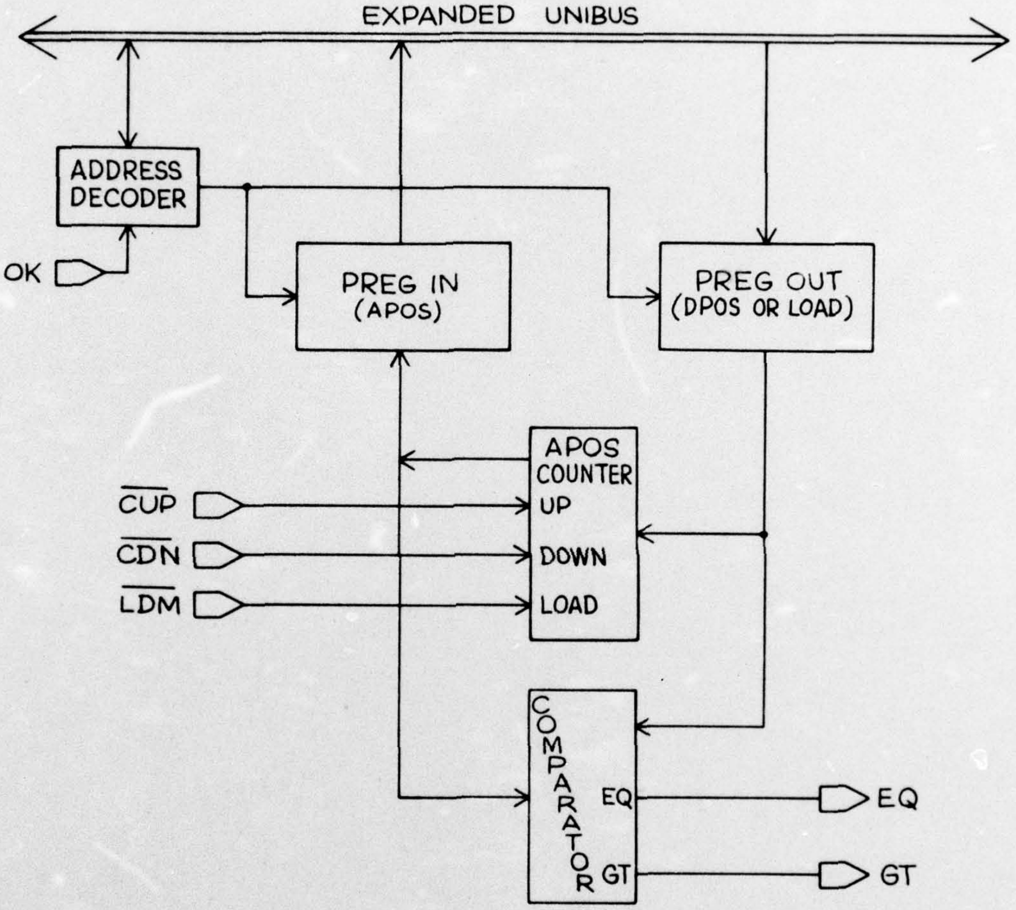
8.10. Position Register - Figure 16

The Position Register (PREG) is actually an Expanded Unibus General Register Card with a Counter/Comparator addition. The output side of the register feeds one set of comparator inputs and may also be loaded into the 16-bit counter which feeds the other set of comparator inputs. Therefore, the comparator outputs, Equal and Greater Than (EQ and GT), pertain to the relation between the counter and the output side of the register. The input side of the register is fed by the counter output. In the step motor controller, the counter contains the value of APOS, and the input side of the register contains the value of DPOS.



STATUS REGISTER- BITS 8-15

Figure 15



POSITION REGISTER (PREG)

Figure 16

9. PAN/TILT USERS MANUAL

This section is intended to serve as a guide to writing sofrware for the pan/tilt controllers. Both controllers appear absolutely identical to the programmer except for their addresses. The terms Status Register (STREG) and Position Register (PREG) apply to either the pan or the tilt controller in the following section.

9.1. Device Addresses

765200	- Pan Status Register
765202	- Pan Position Register
300	- Pan Interrupt Vector
765204	- Tilt Status Register
765206	- Tilt Position Register
304	- Tilt Interrupt Vector

9.2. Status Register

Bit Description

15 - Error. Actually the logical "or" of bits 13 and 14

14 - Plus Limit Error. Indicates an attempt to move an axis beyond its range in the upward or CW direction. Inhibits further steps in that direction.

13 - Minus Limit Error. Indicates an attempt to move an axis beyond its range in the downward or CW direction. Inhibits further steps in that direction.

12 - No Movement (NM). Indicates that there has been no rotor motion within the previous 1/2 second.

11 - Line. Indicates that the Line/Local switch is in the line position, enabling the controller and disabling the joysticks.

10 - Jdone. Indicates that the joystick done switch is depressed. This switch is only a sense switch and has no effect upon the controller.

9 - E1. Current value of shaft encoder 1.

8 - E2. Current value of shaft encoder 2.

7 - Done. Done is set by conditions which are determined by the current mode. (See bits 0 - 2 below.) Done is reset by writing the low order byte of the Status Register.

6 - Interrupt Enable. Causes the computer to be interrupted by Error (Bit 15), Jdone (Bit 10), or Done (Bit 7). It is cleared by a reset instruction.

5 - Not used.

4 - High order bit of DPHASE in Mode 2. Causes a CW step in Mode 3. Cleared by any step while in Mode 3.

3 - Low order bit of DPHASE in Mode 2. Causes a CCW step in Mode 3. Cleared by any step while in Mode 3.

0-2 - Mode select bits. Cleared by a reset instruction.

0 - Stop Mode (STM). Inhibits all step pulses to the motor driver cards. Done when NM is asserted.

1 - Seek Mode (SKM). Causes the device to seek until its actual position (APOS) is equal to the Position Register. Done when the position is reached and NM is asserted.

2 - Slew Mode (SLM). Causes the device to accelerate by setting DPHASE to the contents of bits 3-4:

00 - No acceleration.
01 - CW or upward acceleration.
10 - Not meaningful.
11 - CCW or downward acceleration.

Obviously, friction will limit speed and cause stopping in the absence of acceleration. Done when APOS is equal to PREG. Note that Done is not dependent upon NM, therefore the device may still be moving.

3 - Single Step Mode (SSM). Causes step pulses to be generated as specified by bits 3 and 4.

Bit 3 = > Step downward or CCW.

Bit 4 = > Step upward or CW.

If both bits are set, step pulses in both directions will be generated, yielding unpredictable results. Bits 3 and 4 are both cleared by step pulses while this mode is set. Done when the half way point between rotor equilibrium points is passed, that is, when the motion detector generates an UP or a DOWN signal.

4 - Load Mode (LDM). The hardware actual position counter (APOS) is loaded with the contents of PREG. Done is not generated.

9.3. Position Register

Writing the PREG address actually causes loading of PREG. Reading the PREG address does not cause reading of PREG! Instead, the number obtained is the current value of APOS. Changing PREG while in Seek Mode will cause the device to seek as usual, but will not reset the Done bit. It is therefore advisable to always enter Stop Mode before writing PREG, unless it is certain that the device is not currently in Seek Mode.

10. POSSIBLE IMPROVEMENTS

The overall performance of the Step Motor Controller for the pan/tilt head has been satisfactory. A step rate of almost 200 steps per second, which corresponds to an axis rotation rate of 2.5 RPM, can be obtained even with the load imposed by the TV camera and laser rangefinder.

Although the discrete motion characteristics of step motors guarantee stable and repeatable rest positions, they also have been the source of some difficulties. At slow step rates, mechanical resonance with the pan/tilt stand sometimes causes severe vibration, precluding the use of the TV camera while in motion. An obvious solution would be to use an ordinary DC motor and brake combination in place of the step motor, while still sensing rotor motion with optical shaft encoders. Proper selection of the DC motor would allow faster slew rates and smoother slow speed operation at the expense of a somewhat more complicated controller. The brake would facilitate maintaining a fixed position, and the optical shaft encoders would provide positional accuracy as good as that in the present system.

Even in such a modified controller, the synchronous multi-phase design approach could still be used to full advantage in decoding the shaft encoder output, performing the control algorith, and interfacing to the controlling computer. As in the current controller, this approach would help organize the control algorithm into sequential steps, allow each step to be reduced to combinational logic design, avoid race conditions, and encourage accurate computer simulation before construction.

APPENDIX A

The following is a MACRO-11 subroutine used to simulate the step motor control algorithm on a PDP-11. The address of the shaft encoder input and step command register, and the desired relative displacement are both passed to the subroutine on the stack. A motion sequence is considered complete when the NM flag comes on after having been off.

```
        .TITLE  STEP MOTOR CONTROLLER SIMULATION
;R0     = CURRENT ENCODER POSITION
;R1     = ACTUAL PHASE (APHASE)
;R2     = PREVIOUS ENCODER POSITION
;R3     = DESIRED PHASE (DPHASE)
;POS    = NUMBER OF STEPS WHICH THE MOTOR IS TO TAKE IN
;         THE CW DIRECTION (NEGATIVE NUMBER INDICATES CCW)
RN1     = 400    ;REGION 1 WHEN E1 AND NOT E2
RN4     = 0      ;REGION 4 WHEN NOT E1 AND NOT  E2
CM      = 176377          ;MASK TO GET ENCODER VALUES
NM      = 10000 ;NM BIT
PLERR   = 40000 ;+ LIMIT ERROR BIT
MIERR   = 20000 ;- LIMIT ERROR BIT

        .GLOBL  MOVE,$SAV5
        .CSECT  CONTR
MOVE:   JSR     R1,$SAV5            ;SAVE REGS
L10:    BIT     #NM,@16.(SP)        ;CHECK NO MOVEMENT FLAG
        BEQ     L10                 ;LOOP UNTIL ASSERTED
        CLR     R1                  ;APHASE IS 0 WHEN STOPPED
        MOV     #-1,R2              ;INIT PREVIOUS POSITION
        MOV     14.(SP),POS         ;GET POS FROM STACK
        MOVB    #1,WMOV             ;SET FLAG TO WAS NOT MOVING
L20:    MOV     @16.(SP),R0         ;GET CURRENT POSITION
        BIT     #NM,R0              ;IS ROTOR MOVING?
        BEQ     L21                 ;YES, SKIP
        TSTB    WMOV                ;ELSE WAS ROTOR MOVING?
        BNE     L22                 ;NO, IT HASN'T STARTED YET
        CLR     R0                  ;ELSE IT HAS STOPPED AND
        RTS     PC                  ;   IT'S TIME TO RETURN
L21:    CLRB    WMOV                ;SET FLAG TO MOVING
L22:    BIC     #CM,R0              ;MASK OFF ENCODER BITS
```

```
        CMP     R2,#RN4         ;WAS PREVIOUS POSITION RN4?
        BEQ     L70             ;YES, BRANCH
        CMP     R2,#RN1         ;WAS PREVIOUS POSITION RN1?
        BEQ     L80             ;YES, BRANCH
L30:    MOV     R0,R2           ;SAVE CURRENT POSITION
        TST     POS             ;SEE IF POS IS 0
        BEQ     L40             ;SKIP IF IT IS
        BPL     L50             ;SKIP IF MORE STEPS TO GO
        MOV     #3,R3           ;POS < 0, SET DPHASE TO -1
        BR      L60
L40:    CLR     R3              ;POS = 0, SET DPHASE TO 0
        BR      L60
L50:    MOV     #1,R3           ;POS > 0, SET DPHASE TO 1
L60:    MOV     R3,R4           ;MOV DPHASE TO R4
        ASL     R4              ;SHIFT LEFT BY 2 BITS
        ASL     R4
        BIS     R1,R4           ;OR IN APHASE
        MOVB    STEP(R4),R4     ;LOOK UP DIRECTION IN TABLE
        BEQ     L20             ;LOOP IF NO STEP
        CMP     R4,#5           ;CW STEP?
        BEQ     L62             ;YES, SKIP
        BIT     #MIERR.@16.(SP) ;ELSE CHECK FOR MINUS LIMIT
        BEQ     L64             ;SKIP IF NO ERROR
        MOV     #-1,R0          ;- LIMIT ERROR
        RTS     PC
L62:    BIT     #PLERR,@16.(SP) ;CHECK FOR PLUS LIMIT
        BEQ     L64             ;SKIP IF NO ERROR
        MOV     #1,R0           ;+ LIMIT ERROR
        RTS     PC              ;RETURN
L64:    MOV     R4,@16.(SP)     ;STEP IN INDICATED DIRECTION
        ADD     R4,R1           ;ADJUST PHASE
        BIC     #177774,R1      ;       MODULO 4
        BR      L20
L70:    CMP     R0,#RN1         ;IS CURRENT POSITION RN1?
        BNE     L30             ;NO, BRANCH
        DEC     POS             ;ELSE HAS MOVED CW
        ADD     #3,R1           ;ADJUST APHASE
        BIC     #177774,R1      ;       MODULO 4
        BR      L30
L80     CMP     R0,#RN4         ;IS CURRENT POSITION RN4?
        BNE     L30             ;NO, BRANCH
        INC     POS             ;ELSE HAS MOVED CCW
        INC     R1              ;ADJUST APHASE
        BIC     #177774,R1      ;       MODULO 4
        BR      L30
```

```
;STEP DIRECTION LOOKUP TABLE:
;       0 = NO STEP
;       5 = STEP FORWARD OR CW
;       7 = STEP BACKWARD OR CCW
STEP:   .BYTE   0,7,5,5
        .BYTE   5,0,7,5
        .BYTE   7,5,0,7
        .BYTE   7,7,5,0


POS:    .WORD                           ;DISPLACEMENT
WMOV:   .BYTE                           ;WAS MOVING FLAG
        .EVEN
        .END
```

REFERENCES

1. Digital Equipment Corp., PDP-11 Peripherals Handbook, Digital Equipment Corp., Maynard, Mass., 1973.

2. Digital Equipment Corp., PDP-11 Processor Handbook, Digital Equipment Corp., Maynard, Mass., 1972.

3. Geschke, C. C., Jacobus, C. J., and Williamson, D. A., "PDP-11 Unibus Expander and General Interface," Advanced Automation Research Group, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois, 1975.

4. Kuo, B. C., Yamanaka, J. C., Yackel, R., and Singh, G., "Design of a Closed-Loop Controller for a Step Motor," CSRL Report No. 218, Control Systems Research Laboratory, University of Illinois, Urbana, Illinois, August 1972.